SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2026, Vol. 69, Iss. 8, 182102:1–182102:10
https://doi.org/10.1007/s11432-024-4881-y

Motion-driven 4D scene generation

Guo-Wei YANG[1,2], Qun-Ce XU[1*], Zhao WEI[3] & Shi-Min HU[1]

[1]*Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*
[2]*Fitten Tech Co., Ltd., Beijing 100084, China*
[3]*Tencent Tech Co., Ltd., Shenzhen 518000, China*

Received 24 December 2024/Revised 2 April 2025/Accepted 8 May 2025/Published online 31 July 2026

Abstract Recent advances in image/video generative models and learnable dynamic scene representations have enabled the easy creation of 4D scenes. However, the question of how to achieve more controlled motion generation rather than providing high-level text prompts is not well explored. This paper presents an innovative method that leverages user-specified action paths to guide the 4D scene generation. The challenges of motion-controlled 4D scene generation mainly lie in the temporal variations of the motion and the high requirement for temporal consistency. To address these, our method incorporates an adaptive motion alignment module to enhance the naturalness of the generated motion, which dynamically synchronizes motions in the action path domain with their corresponding contents in the time domain. Additionally, to mitigate artifacts caused by excessive motion, we introduce a progressive motion drive module to ensure high-quality outputs by enhancing the control of motion. Experimental results demonstrate the superiority of our method in performing motion-controlled 4D scene generation.

Keywords 4D generation, dynamic 3D scene, diffusion model, motion control, temporal consistency

Citation Yang G-W, Xu Q-C, Wei Z, et al. Motion-driven 4D scene generation. Sci China Inf Sci, 2026, 69(8): 182102, https://doi.org/10.1007/s11432-024-4881-y

1 Introduction

Generative models have attracted significant research attention in the field, aiming to utilize machine learning models to generate realistic data with different dimensions, such as audio, images, and 3D models [1–6]. Images are the most commonly used and extensively researched data format among these studies. In particular, with the assistance of diffusion models, researchers can now create controlled and lifelike images [7, 8]. The advancements in mature image generation models have undoubtedly propelled research in video generation. Researchers have attempted to advance video generation through approaches based on 2D images. Many video modeling methods draw inspiration from techniques in the image domain, often training on mixed image and video datasets and incorporating additional temporal layers for fine-tuning using pre-trained models [9, 10].

However, only learning from 2D data to generate videos of dynamic scenes has drawbacks. First, they result in inaccurate object positions and distances due to the lack of depth information. Furthermore, the lack of precise object shapes and structures in the current approach results in the presence of blurry artifacts, which significantly degrade the visual quality of the generated videos. Also, limited viewpoints restrict perspective diversity, hindering a comprehensive scene understanding. Furthermore, the results only represent partial scenes, lacking information about occluded objects and backface details. These limitations can be addressed by incorporating a 3D learnable representation during the generation process and bridging the dimensionality gap using differentiable rendering.

Building upon previous work in generating 3D scenes [11–14], we aim to address the limitations of 2D data by incorporating their underlying 3D scenes as prior knowledge. Leveraging the scene consistency provided by 3D data can help mitigate artifacts resulting from the lack of information in 2D. Also, we want the generated results to be more controllable beyond the textual level. However, achieving precise control over 4D scenes while maintaining quality results poses challenges due to the high demands of temporal and spatial consistencies in 3D scenes.

To tackle these challenges, we propose a motion-driven 4D scene generation approach. Our approach allows the use of action paths to give detailed control of the generated motion (see Figure 1). We introduce an adaptive motion alignment module (AMAM), which aligns the variations in the 3D scene with the motion path features in the latent space, resulting in smooth and natural transitions along the motion path. Additionally, to address

* Corresponding author (email: quncexu@tsinghua.edu.cn)

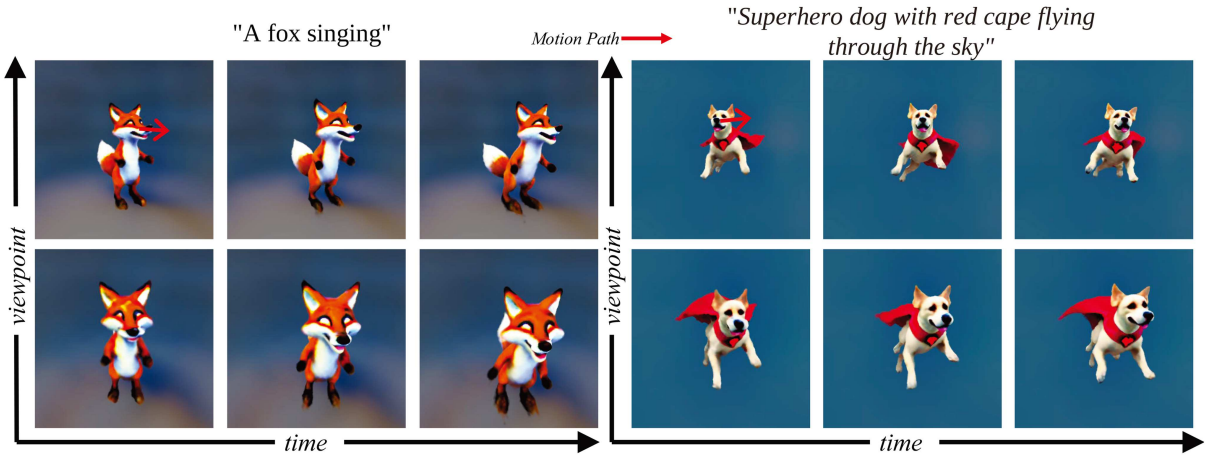

Figure 1 We presented a motion-driven method that can generate dynamic 3D scenes from a text prompt with a given action path (red arrow) that guides the generated motion.

potential artifacts caused by significant variations in motion magnitude, we propose a progressive motion drive module (PMDM) that gradually changes the motion. It effectively avoids the aforementioned issue and ensures the quality of the generated results. We extensively evaluate our work using a variety of generated videos based on different text prompts and action path controls. Both qualitative and quantitative comparisons and ablation studies demonstrate the advantages of our work over baselines.

In summary, our work makes the following main contributions.

• We propose a motion-driven 4D scene generation model that addresses challenges in consistency and artifacts in motion-controlled 4D scene generation through the adaptive motion alignment module (AMAM) and progressive motion drive module (PMDM).

• We introduce the cascade hypercube representation and motion-driven loss, which allow precise control over the generated 4D scenes.

• We conduct quantitative and qualitative evaluations to demonstrate the effectiveness and superiority of our model.

2 Related work

2.1 Video generation

Recently, numerous studies on video generation have emerged. LVDM [15] performs video generation based on the latent diffusion model. AnimateDiff [9], MAV [16], and SVD [17] extend the text-to-image (T2I) model to text-to-video (T2V) model, taking advantage of the significantly larger amount of image data than video. VideoFactory [18], VideoFusion [19], and Lumiere [20] enhance the temporal consistency of the video by exchanging inter-frame information. VideoComposer [21] and MotionCtrl [22] provide additional inputs for the camera motion as well as the motion of the video content. Animate Anyone [23] generates videos with controlled character movements based on human pose sequences. Based on StyleGAN [4], DragGAN [24] controls image changes by dragging to generate image sequences in real-time interactively. Other works like W.A.L.T. [25] propose to generate videos through a Diffusion model based on Transformers. Commercial models such as Sora [26], Runway [27], and Pika [28] are trained based on vast amounts of data and achieve high-quality video generation results. However, the videos generated by existing methods still have problems such as 3D inconsistencies and non-conformity to physical laws, which may be because the existing generation methods are mainly based on 2D content generation without explicitly generating 3D scenes.

2.2 3D generation

With the advancement of generative models, 3D content generation has become a highly active research field. Building upon the success of generative models such as GANs and VAEs in 2D, methods for generating 3D content [29–33] have gradually emerged. However, due to the scarcity of 3D data and the difficulties in model training, early efforts in 3D generation are often confined to generating a single category of objects, such as faces or vehicles. Recently, novel learnable 3D representations like NeRF [34] and triplane [35] have made significant changes to

traditional approaches. Similar to the progress in 2D content generation, the popularity of diffusion models has also driven research in 3D scene generation, leading to numerous promising works. Some works utilize diffusion models to learn and generate 3D models, such as Shap-e [36]. Other approaches, like [35] and [37], combine triplane representations with diffusion models to generate high-quality 3D scenes. On the other hand, Point-E [12] combines the point cloud representation with diffusion models to generate 3D point clouds. Meanwhile, based on other traditional representations, MeshDiffusion [38] and 3D-LDM [39] are concurrently proposed.

However, the scarcity of 3D datasets and the required computational resources make direct training of 3D diffusion models extremely costly. To address this challenge, some works in the diffusion model-based 3D content generation field have shifted their focus to image-based training. One pioneering work in this area is Zero-1-to-3 [40], which uses a diffusion model and CLIP [41] to generate 3D models. This method enables flexible control over the generated content attributes through natural language prompts without the need for any 3D training data. Another notable work, DreamFusion [13], also leverages text-to-image diffusion models Imagen [42] to generate 3D shapes while combining the concepts from Mip-NeRF [43] and employing the score distillation sampling (SDS) loss. These works bridge the gap between text and 3D models by utilizing images and the NeRF representation. Subsequently, Magic3D [11] improves the efficiency of 3D content generation by replacing Mip-NeRF with Instant-NGP [44] as the representation. Similarly, Make-it-3D [45] generates high-quality 3D models from a single image using a two-stage framework based on the NeRF representation. It optimizes NeRF and incorporates high-quality textures from the input image and diffusion priors, enhancing the quality of the generated textures.

2.3 4D scene generation

While researchers have been exploring the generation of static 3D scenes, how to represent and generate 4D dynamic scenes has also received significant research attention. NeRFies [46] combines deformation fields with regularization methods to adjust the deformation field of the scene and simulate high-frequency variations. D-NeRF [47] employs decomposition learning of canonical field and scene flow to represent space and time and render dynamic scenes. Similar research endeavors, such as [48, 49], have also explored how to incorporate temporal information. Ref. [50] introduces depth loss as supervision based on scene depth to constrain the temporal geometry of dynamic scenes, resulting in dynamic video generation with free viewpoints. Refs. [51, 52] employ scene flow to obtain dynamic representations in NeRF. In addition, as an extension of triplanes [35], HexPlane [53] employs six planes to represent the feature space of 4D scenes. On the other hand, K-Planes [54] naturally divides the scene into static and dynamic components through plane decomposition. Both of them effectively reduce storage and computational complexity. Recently, there have been notable advancements in 4D scene rendering using Gaussian splatting [55], as exemplified in [56, 57].

MAV3D [58] is the first to generate 4D scenes based on textual prompts by expanding the generation model into 4D space. Given input texts, MAV3D achieves consistency in appearance, density, and motion by optimizing dynamic NeRF-based scenes using T2V diffusion models [16] and temporal score distillation sampling (SDS-T). Further works, such as 4D-fy [59], which introduces hyper SDS, and AYG [60], which uses 3D Gaussians to represent 4D scenes, continue to contribute to the text-based 4D scene generation field. Another notable work, Control4D [61], utilizes GaussianPlanes as the 4D representation and ControlNet [62] as the image editor to edit dynamic 4D content generation with textual prompts. Despite the achievements of the works discussed above for generating dynamic 3D scenes, challenges still exist in controlling and editing motions in the generated 4D scenes that we aim to address in this work.

3 Motion driven text-to-4D generation

Figure 2 shows the pipeline of our method. In order to make different locations in space decoupled from each other, we adopt cascade hypercube representation for 4D scene representation (Subsection 3.1). To achieve action-driven 4D scene generation, we propose the AMAM, which adaptively aligns the motion in the domain of the action path to match the corresponding parts of the action in the time domain (Subsection 3.2). Also, large-scale motion can lead to significant artifacts during generation. To address this issue, we propose incrementally enhancing the motion drive through the PMDM, ensuring the generation result's quality (Subsection 3.3).

3.1 Cascade hypercube representation

Previous 4D scene generation methods, such as MAV3D [58], often use HexPlane [53] as the feature representation of the dynamic scene. For our motion-driven 4D scene generation task, we need to carefully control the corresponding

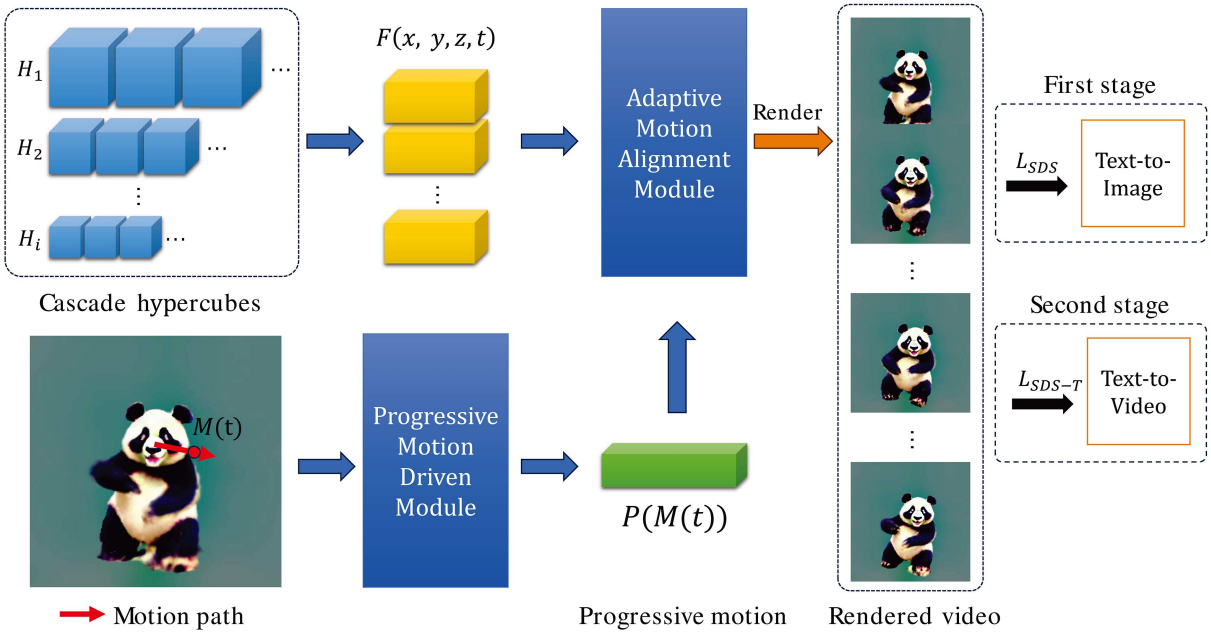

Figure 2 Pipeline of our method. We employ the cascade hypercube representation to represent 4D scenes. Utilizing the AMAM, we achieve domain-adaptive alignment of action features along the action paths, enabling matching action correspondences in the temporal domain to drive motion. The PMDM is employed to progressively enhance motion driving, addressing issues related to significant artifacts during generation and training instability that may arise from extensive motion.

Figure 3 Generation results based on HexPlane representation. The first two columns depict rendering results at different times from a particular viewpoint, while the latter two columns show rendered results from another viewpoint at the same time stamp. The red arrow in the first column of images indicates the action path for motion condition.

part of the scene space according to the given action path. However, modifying any point in one feature plane of the HexPlane-based feature representation will affect the other feature dimensions. For example, F_{XY} represents a special HexPlane feature in the XY plane, and modifying any point of F_{XY} will have an impact on the spatial Z-axis and the temporal T-axis, which will cause a severe perturbation to the scene, thus leading to the failure of the training as it cannot converge correctly. Figure 3 shows the effect of using HexPlane for motion-driven 4D scene generation. It can be seen that HexPlane-based generation results in clearly visible artifacts.

Therefore, we employ the 4D scene feature representation by cascade hypercube to ensure that the features are decoupled from each other in different dimensions. For a point (x, y, z, t) in the 4D space, its features can be obtained by hypercube stacks with different resolutions:

$$F(x, y, z, t) = [H_1(x, y, z, t); \ldots; H_C(x, y, z, t)], \tag{1}$$

where C is the number of hypercubes at different resolutions and $H_i(x, y, z, t)$ denotes the hypercube feature at the i-th resolution at point (x, y, z, t).

3.2 Adaptive motion alignment module

We expect the generated 4D scene to satisfy the constraints of our action in specific parts. We define an action M in 3D space as a keypoint path that can be defined by the user. $M(t)$ denotes the coordinates of the keypoint at

time t. We measure the fitness of the 4D scene concerning the driven motion as follows. First, we sample points near the critical points at different time stamps. Then, we define the motion fitness of the 4D scene based on the color and density of the sampled points as

$$L_m(M) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\Big\|R(M(0)+p_i,0) - R(M(t)+p_i,t)\Big\|_1, \tag{2}$$

where L_m is motion loss, N is the number of sample points, p_i is one of the randomly sampled points around the origin, T is the number of frames in the output video, and $R(p,t)$ denotes the color and density at the spatial location p at time t based on the hypercube 4D scene representation.

Simply performing motion supervision via L_m restricts the relative position of the motion keypoint domain to be invariant, which is likely to be inconsistent with the regularity of the motion, thus leading to the rigidity of the generated motion. Figure 4 illustrates the motion-driven generation effect using only L_m, which has an apparent unnatural head distortion. Therefore, we propose the AMAM, which can adaptively adjust the local displacement of the keypoints during the training process such that they can be naturally aligned at different times of the motion. The AMAM is also based on the hypercube representation, which takes as input the local spatial coordinates (x, y, z) along with time t and returns an adaptive alignment offset at time t.

$$(\mathrm{d}x, \mathrm{d}y, \mathrm{d}z) = \mathrm{A}(x, y, z, t), \tag{3}$$

where A is the AMAM and $(\mathrm{d}x, \mathrm{d}y, \mathrm{d}z)$ is the adaptive alignment offset of coordinates (x, y, z) at time t.

Based on the AMAM, we define the adaptive motion loss L_{am} as

$$L_{am}(M)=\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\|R(M(0)+p_i,0)-R(M(t)+p_i+\mathrm{A}(p_i,t),t)\|_1. \tag{4}$$

To prevent the AMAM from overfitting to the point with no changes in motion, thus making the motion ineffective, we also employ a regularization loss which ensures that the average offset of the sampling points is close to 0:

$$L_R = \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\|\mathrm{A}(p_i,t)\|_2. \tag{5}$$

3.3 Progressive motion drive module

When the motion amplitude is large, the direct motion drive will generate artifacts due to the significant change of content at the time stamp of the large motion. Therefore, we propose incrementally enhancing the motion intensity based on the PMDM so that the resultant 4D scene can gradually transform into a scene that meets the motion approximation. We enhance the motion intensity in multiple stages during training.

First, we define the progressive motion $P(M)$ for motion M as

$$P(M(t)) = M(0) + \frac{stage(iter)}{S}M(t), \tag{6}$$

where S is the total number of stages, $iter$ is the current number of training rounds, and $stage(iter)$ is the number of stages the current iter belongs to.

$$\begin{aligned} stage(iter) &= \left\lceil \frac{iter}{s} \right\rceil, \\ S &= \left\lceil \frac{U}{s} \right\rceil. \end{aligned} \tag{7}$$

The PMDM enhances the amplitude of the motion path progressively. Specifically, it starts from an initial amplitude of 0 and increases the amplitude of the path every s rounds, forming a new stage. This process continues until the final stage, where the generated path matches the user's input path exactly.

Here, s represents the number of training rounds in each stage, U is the total number of training rounds for the PMDM, and S is the total number of stages, calculated as $S = \lceil \frac{U}{s} \rceil$. The motion amplitude is scaled by the factor $\frac{\mathrm{stage}(iter)}{S}$, which increases progressively from 0 to 1 as training moves from the first stage to the final stage.

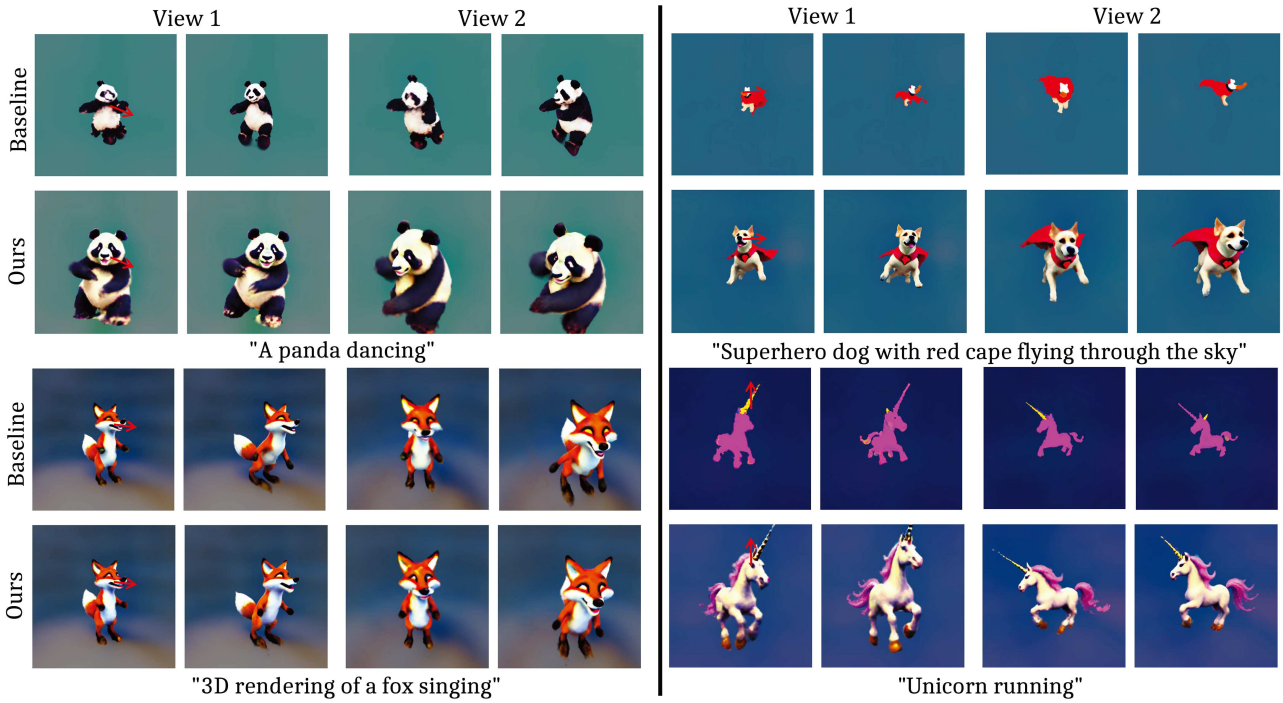

Figure 4 Visual comparison between the results generated by our method and those of the baseline. The baseline result of "superhero dog with red cape flying through the sky" is a broken case due to training failure.

From the above, the adaptive motion loss based on the PMDM can be defined as

$$L_{am}(P(M)) = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\|R(P(M(0))+p_i,0) - R(P(M(t))+p_i+\mathrm{A}(p_i,t),t)\|_1. \tag{8}$$

We adopt a multi-stage training approach similar to MAV3D [58]. In the first stage, we trained static scenes using the L_{SDS} based on the text-to-image diffusion model. In the second stage, we enhanced the three-dimensional consistency of static scenes through the $L_{\mathrm{SDS\text{-}T}}$ based on the text-to-video diffusion model. In the third stage, we utilized $L_{\mathrm{SDS\text{-}T}}$ and our motion-driven loss.

$$L = \alpha_1 L_{\mathrm{SDS\text{-}T}} + \alpha_2 L_{am}(P(M)) + \alpha_3 L_R, \tag{9}$$

where α_1, α_2, and α_3 are the weights for $L_{\mathrm{SDS\text{-}T}}$, $L_{am}(P(M))$, and L_R, respectively.

4 Experiments

4.1 Metrics

We evaluate the quality of the generated results by measuring the cosine similarity between the CLIP [41] features of the ViT-B/32 model extracted from the rendered images and the text prompt, which is called CLIP score [63]. We compute the average CLIP score over 16 frames rendered from two perspectives for each prompt. Besides, we conducted a user study to evaluate our method and baseline for seven different prompts. We collected human preferences based on video quality, motion quality (w.r.t. the guided path), and realistic motion scores at 1–7 different levels.

4.2 Implementation details

We utilized the pre-trained stable diffusion [64] and the text-to-video model provided by 4D-fy [59] on Hugging Face. We conducted training with 2000 iterations in the first stage, 3000 iterations in the second stage, and 10000 to 20000 iterations in the third stage (depending on the convergence of the model). The batch size during training was set to 1. In the initial two stages of training and the first 4000 iterations of the third stage, the rendered image

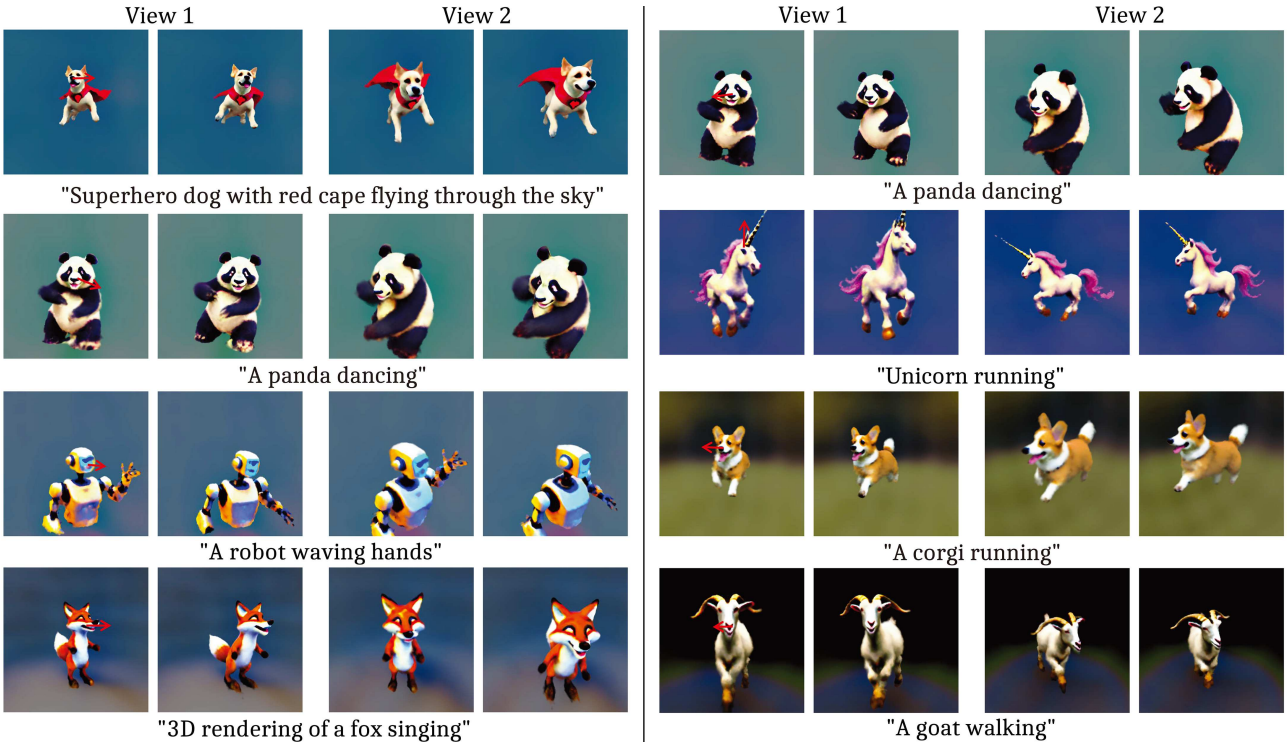

Figure 5 Generated results based on our method and the corresponding prompts. In each set of results, the first two columns depict rendering outcomes at different times from a particular viewpoint, while the last two columns show rendering results from another viewpoint at the same time stamp. The red arrows in the first column indicate the action paths for motion conditions.

resolution was 64×64 pixels. After 3500 iterations in the third stage, we switched to an image resolution of 128×128 pixels. In the last two stages, we rendered 16 frames of video for training.

4.3 Results and comparisons

Figures 1 and 5 showcase the generation results of our method across multiple prompts and motion scenarios. It can be observed that the results demonstrate high-quality and natural motion-driven effects. For instance, in the result for "superhero dog with red cape flying through the sky", the motion drives the dog's head to move to the right, and its feet naturally swing to the left. MAV3D [58] and other 4D generation works are based on HexPlane as a scene representation. Figure 3 illustrates the generated results based on HexPlane scene representation, revealing its inability to produce reasonable outcomes. Noticeable artifacts, particularly pronounced artifact shadows, emerge in the motion-driven sections. This can be attributed to the significant coupling between different dimensions in this representation, disrupting the scene during motion-driven processes and consequently hindering the generation of plausible results.

Due to the limitation of the HexPlane-based method that fails to accurately perform motion-driven processes, we adopt the cascade hypercube for scene representation. Additionally, we incorporate motion supervision through motion loss in (2). Figure 4 compares its generated results and ours. It can be observed that in the results for "a panda dancing", due to the excessively large motion amplitude, the method fails to drive the panda's head to move, resulting in artifacts at the target points of motion. In the results for "superhero dog with red cape flying through the sky", the instability introduced by the driving motion makes the scene fail to converge during training. Table 1 presents the CLIP score between our method and the baseline method, demonstrating that our approach outperforms the baseline method. Also, the collected human preferences show that our method achieves higher scores than the baseline in video quality, motion quality, and realistic motion. This validates the effectiveness of our method for generating high-quality dynamic 3D scenes under motion control.

To demonstrate the versatility of our method, we also utilized the VideoFusion [19] code on Hugging Face as our backbone network. With the assistance of our proposed module, it was able to successfully generate videos corresponding to user-specified motion paths from text. Figure 6 illustrates the qualitative results generated. Due to the inherent performance differences of the backbone networks, the results from 4D-fy [59] are better than those from VideoFusion [19].

Table 1 Quantitative comparison results. Bold numbers indicate the best. Here are the CLIP score and human preference study between the baseline and our method. CS stands for CLIP score, VQ for video quality, MQ for motion quality, and RM for realistic motion.

Method	CS$^{\uparrow}$	VQ$^{\uparrow}$	MQ$^{\uparrow}$	RM$^{\uparrow}$	Overall$^{\uparrow}$
Baseline	29.0569	2.71	3.02	2.75	2.83
Ours	**32.0278**	**4.64**	**4.79**	**4.64**	**4.69**

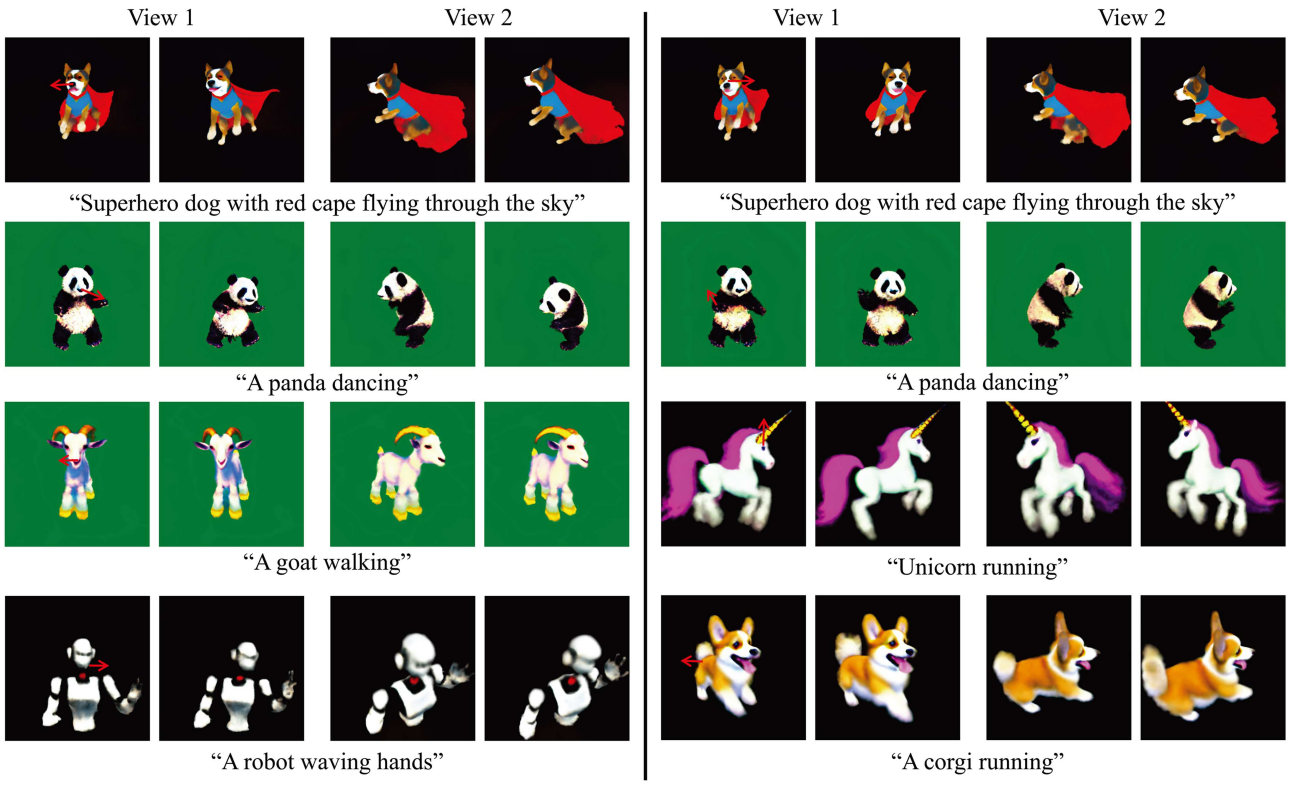

Figure 6 To demonstrate the ease of generalization of our method to different backbone networks, we replaced the backbone network with VideoFusion [19] and generated visual results.

4.4 Ablation study

Figure 7 shows the visual comparison between our method and the ablative method, while Table 2 presents the quantitative comparison based on CLIP score. Table 2 reports the CLIP scores across seven prompts. Our full model achieves the highest average score of 32.08, confirming the effectiveness of both proposed modules. The ablative experiments conducted on the AMAM demonstrate that lacking this module cannot adaptively adjust the local alignment of actions. This leads to unnatural motion transitions and loss of details compared with the complete method. Introducing the AMAM ensures smooth transitions between actions and accurate representation of details, significantly enhancing the realism and naturalness of the generated dynamic scenes.

We also compare our method with a version that does not utilize the PMDM. The results of the ablation experiments show that the lack of PMDM cannot accurately generate dynamic 3D scenes. They may produce severe artifacts at the target points of the action, fail to drive the scene with the expected action, and lead to training instability that causes scene crashes. In contrast, introducing the PMDM allows our method to generate plausible dynamic scenes driven by actions, ensuring the coherence of actions and the overall coordination of scene generation.

These ablation studies demonstrate the crucial roles of the AMAM and the PMDM in our method. The AMAM enhances the naturalness and accuracy of dynamic 3D scenes by intelligently adjusting the local alignment of actions. The PMDM ensures the coherence of actions and successful scene generation, significantly improving the overall quality of the generated scenes while avoiding unexpected artifacts.

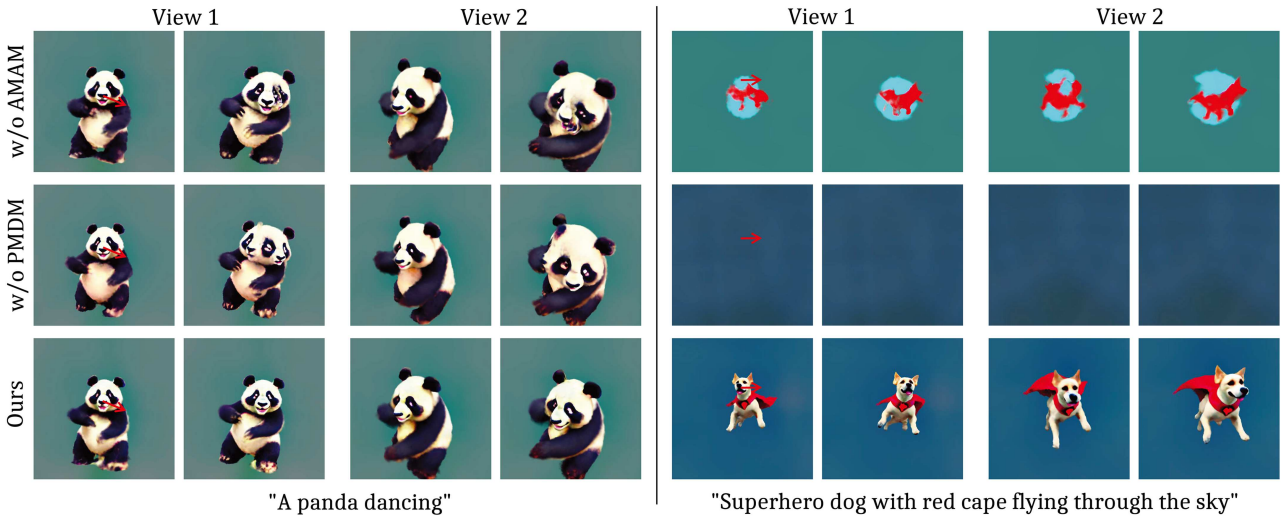

Figure 7 Visual comparison between the results generated by our method and the ablative methods.

Table 2 CLIP scores of our method and the ablative methods for generating scenes. Bold numbers denote the best results.

Method	Corgi$^{\uparrow}$	Fox$^{\uparrow}$	Goat$^{\uparrow}$	Panda$^{\uparrow}$	Robot$^{\uparrow}$	Superhero$^{\uparrow}$	Unicorn$^{\uparrow}$	Average$^{\uparrow}$
w/o AMAM	34.1474	28.6662	31.4214	30.2373	28.2583	24.9392	30.5870	29.7509
w/o PMDM	33.9221	28.7867	21.0898	30.3244	28.2249	20.8409	30.4635	27.6646
Ours	**35.8813**	**29.8065**	**31.6283**	**30.8637**	**28.9476**	**36.2810**	**31.1602**	**32.0812**

5 Conclusion

This paper introduces a motion-driven method for dynamic 3D scene generation, which allows the control of 4D generation using action paths. Our method leverages the AMAM to constrain the generated scenes to align with motion conditions, enhancing the naturalness of the generated results. Additionally, we proposed the PMDM to incrementally enhance the driven motion, addressing issues related to artifacts caused by large motions. The experiments conducted have demonstrated the effectiveness of our proposed method. The results indicate that, in comparison to the baseline, our approach achieves higher-quality outcomes.

However, this work still has limitations in terms of the diversity of the motion, as well as the duration and clarity of the generated dynamic scenes. In future work, we plan to enrich the diversity of action paths by leveraging motion capture data and pre-trained motion models, thereby generating more complex and realistic motions. Furthermore, we will enhance training efficiency using transfer learning to reduce the required number of iterations. Additionally, we aim to improve the resolution and detail of the generated scenes through multi-scale training strategies.

References

1 Kingma D P, Welling M. Auto-encoding variational bayes. ArXiv:1312.6114
2 Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: Proceedings of Advances in Neural Information Processing Systems, 2014
3 Sohl-Dickstein J, Weiss E, Maheswaranathan N, et al. Deep unsupervised learning using nonequilibrium thermodynamics. In: Proceedings of International Conference on Machine Learning, 2015. 2256–2265
4 Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 4401–4410
5 Song Y, Sohl-Dickstein J, Kingma D P, et al. Score-based generative modeling through stochastic differential equations. ArXiv:2011.13456
6 Van Den Oord A, Kalchbrenner N, Kavukcuoglu K. Pixel recurrent neural networks. In: Proceedings of International Conference on Machine Learning, 2016. 1747–1756
7 Karras T, Laine S, Aittala M, et al. Analyzing and improving the image quality of styleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 8110–8119
8 Rombach R, Blattmann A, Lorenz D, et al. High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 10684–10695
9 Guo Y, Yang C, Rao A, et al. AnimateDiff: animate your personalized text-to-image diffusion models without specific tuning. ArXiv:2307.04725
10 Wu J Z, Ge Y, Wang X, et al. Tune-a-video: one-shot tuning of image diffusion models for text-to-video generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 7623–7633
11 Lin C H, Gao J, Tang L, et al. Magic3D: high-resolution text-to-3D content creation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 300–309
12 Nichol A, Jun H, Dhariwal P, et al. Point-E: a system for generating 3D point clouds from complex prompts. ArXiv:2212.08751
13 Poole B, Jain A, Barron J T, et al. DreamFusion: text-to-3D using 2D diffusion. ArXiv:2209.14988
14 Wang Z, Lu C, Wang Y, et al. ProlificDreamer: high-fidelity and diverse text-to-3D generation with variational score distillation. In: Proceedings of Advances in Neural Information Processing Systems, 2024

15 He Y, Yang T, Zhang Y, et al. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. ArXiv:2211.13221
16 Singer U, Polyak A, Hayes T, et al. Make-a-video: text-to-video generation without text-video data. ArXiv:2209.14792
17 Blattmann A, Dockhorn T, Kulal S, et al. Stable video diffusion: scaling latent video diffusion models to large datasets. ArXiv:2311.15127
18 Wang W, Yang H, Tuo Z, et al. VideoFactory: swap attention in spatiotemporal diffusions for text-to-video generation. ArXiv:2305.10874
19 Luo Z, Chen D, Zhang Y, et al. VideoFusion: decomposed diffusion models for high-quality video generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 10209–10218
20 Bar-Tal O, Chefer H, Tov O, et al. Lumiere: a space-time diffusion model for video generation. ArXiv:2401.12945
21 Wang X, Yuan H, Zhang S, et al. VideoComposer: compositional video synthesis with motion controllability. In: Proceedings of Advances in Neural Information Processing Systems, 2024
22 Wang Z, Yuan Z, Wang X, et al. MotionCtrl: a unified and flexible motion controller for video generation. ArXiv:2312.03641
23 Hu L, Gao X, Zhang P, et al. Animate anyone: consistent and controllable image-to-video synthesis for character animation. ArXiv:2311.17117
24 Pan X, Tewari A, Leimkühler T, et al. Drag your GAN: interactive point-based manipulation on the generative image manifold. In: Proceedings of ACM SIGGRAPH 2023 Conference Proceedings, 2023. 1–11
25 Gupta A, Yu L, Sohn K, et al. Photorealistic video generation with diffusion models. ArXiv:2312.06662
26 Brooks T, Peebles B, Homes C, et al. Video generation models as world simulators. 2024. https://openai.com/research/video-generation-models-as-world-simulators
27 AI R. Gen-2: the next step forward for generative AI. 2023. https://research.runwayml.com/gen2
28 Lab P. Pika. 2023. https://pika.art/
29 Wu J, Zhang C, Xue T, et al. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In: Proceedings of Advances in Neural Information Processing Systems, 2016
30 Nguyen-Phuoc T, Li C, Theis L, et al. Hologan: unsupervised learning of 3D representations from natural images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 7588–7597
31 Gao L, Yang J, Wu T, et al. SDM-NET: deep generative network for structured deformable mesh. ACM Trans Graph, 2019, 38: 1–15
32 Chen W, Ling H, Gao J, et al. Learning to predict 3D objects with an interpolation-based differentiable renderer. In: Proceedings of Advances in Neural Information Processing Systems, 2019
33 Shen T, Gao J, Yin K, et al. Deep marching tetrahedra: a hybrid representation for high-resolution 3D shape synthesis. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 6087–6101
34 Mildenhall B, Srinivasan P P, Tancik M, et al. NeRF. Commun ACM, 2021, 65: 99–106
35 Gupta A, Xiong W, Nie Y, et al. 3DGen: triplane latent diffusion for textured mesh generation. ArXiv:2303.05371
36 Jun H, Nichol A. Shap-e: generating conditional 3D implicit functions. ArXiv:2305.02463
37 Shue J R, Chan E R, Po R, et al. 3D neural field generation using triplane diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 20875–20886
38 Liu Z, Feng Y, Black M J, et al. MeshDiffusion: score-based generative 3D mesh modeling. ArXiv:2303.08133
39 Nam G, Khlifi M, Rodriguez A, et al. 3D-LDM: neural implicit 3D shape generation with latent diffusion models. ArXiv:2212.00842
40 Liu R, Wu R, Van Hoorick B, et al. Zero-1-to-3: zero-shot one image to 3D object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 9298–9309
41 Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision. In: Proceedings of International Conference on Machine Learning, 2021. 8748–8763
42 Saharia C, Chan W, Saxena S, et al. Photorealistic text-to-image diffusion models with deep language understanding. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 36479–36494
43 Barron J T, Mildenhall B, Tancik M, et al. Mip-NeRF: a multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 5855–5864
44 Müller T, Evans A, Schied C, et al. Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans Graph, 2022, 41: 1–15
45 Tang J, Wang T, Zhang B, et al. Make-it-3D: high-fidelity 3D creation from a single image with diffusion prior. ArXiv:2303.14184
46 Park K, Sinha U, Barron J T, et al. Nerfies: deformable neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 5865–5874
47 Pumarola A, Corona E, Pons-Moll G, et al. D-NeRF: neural radiance fields for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 10318–10327
48 Tretschk E, Tewari A, Golyanik V, et al. Non-rigid neural radiance fields: reconstruction and novel view synthesis of a dynamic scene from monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 12959–12970
49 Park K, Sinha U, Hedman P, et al. HyperNeRF: a higher-dimensional representation for topologically varying neural radiance fields. ArXiv:2106.13228
50 Xian W, Huang J B, Kopf J, et al. Space-time neural irradiance fields for free-viewpoint video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 9421–9431
51 Gao C, Saraf A, Kopf J, et al. Dynamic view synthesis from dynamic monocular video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 5712–5721
52 Du Y, Zhang Y, Yu H X, et al. Neural radiance flow for 4D view synthesis and video processing. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021. 14304–14314
53 Cao A, Johnson J. Hexplane: a fast representation for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 130–141
54 Fridovich-Keil S, Meanti G, Warburg F R, et al. K-planes: explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 12479–12488
55 Kerbl B, Kopanas G, Leimkühler T, et al. 3D Gaussian splatting for real-time radiance field rendering. ACM Trans Graph, 2023, 42: 1–14
56 Wu G, Yi T, Fang J, et al. 4D Gaussian splatting for real-time dynamic scene rendering. ArXiv:2310.08528
57 Luiten J, Kopanas G, Leibe B, et al. Dynamic 3D Gaussians: tracking by persistent dynamic view synthesis. ArXiv:2308.09713
58 Singer U, Sheynin S, Polyak A, et al. Text-to-4D dynamic scene generation. ArXiv:2301.11280
59 Bahmani S, Skorokhodov I, Rong V, et al. 4D-fy: text-to-4D generation using hybrid score distillation sampling. ArXiv:2311.17984
60 Ling H, Kim S W, Torralba A, et al. Align your Gaussians: text-to-4D with dynamic 3D Gaussians and composed diffusion models. ArXiv:2312.13763
61 Shao R Z, Sun J X, Peng C, et al. Control4D: efficient 4D portrait editing with text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024. 4556–4567
62 Zhang L, Rao A, Agrawala M. Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 3836–3847
63 Zhu X, Sun P, Wang C, et al. A contrastive compositional benchmark for text-to-image synthesis: a study with unified text-to-image fidelity metrics. CoRR, 2023, abs/2312.02338
64 Stability. Stable diffusion v1.5 model card. 2022. https://huggingface.co/runwayml/stable-diffusion-v1-5